Markov Monitoring with Unknown States

Padhraic Smyth
Jet Propulsion Laboratory 238-420
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109.

September 16, 1993

Abstract

Pattern recognition methods and hidden Markov models can be effective tools for on-line health monitoring of communications systems. Previous work has assumed that the states in the system model are exhaustive. This can be a significant drawback in real-world fault monitoring applications where it is difficult if not impossible to model all the possible fault states of the system in advance. In this paper a method is described for extending the Markov monitoring approach to allow for unknown or novel states which can not be accounted for when the model is being designed, The method is described and evaluated on data from one of the Jet Propulsion Laboratory's Deep Space Network antennas. The experimental results indicate that the method is both practical and effective, allowing both discrimination between known states and detection of previously unknown fault conditions.

1 Motivation

Online monitoring of complex communication systems and networks for the purposes of automated fault detection is a topic of considerable practical importance. For example, the Deep Space Network (DSN) (operated by the Jet Propulsion Laboratory for the National Aeronautics and Space Administration (NASA)) consists of three ground-based antenna sites in Australia, Spain, and California. Each site contains a single 70m antenna and several smaller 26m and 34m antennas. These antennas provide the capability for 24-hour communications with various unmanned interplanetary robotic explorers such as the Pioneer, the Voyager, and Magellan spacecraft [1]. Fault detection and isolation is often a complicated and lengthy process due to the fact that it can be difficult to establish the root cause of a problem in the communications chain. As an example, a problem with the radio receiver dropping carrier lock could be caused by a variety of factors such as problems with the spacecraft itself, external environmental influences such as weather, irregularities in the antenna pointing system, problems with the receiver hardware, incorrectly set tracking parameters, and so forth.

Loss of a spacecraft signal during a planetary encounter may result in the irretrievable loss of science data. Hence, there is considerable motivation to be able to quickly detect, isolate, and repair a fault. The same scenario applies to many commercial and military communication systems: rapid detection of failures is essential as communications equipment becomes more complex and various applications (such as medical imaging and electronic financial transactions) have come to rely on communications as an essential component of their day-to-day business.

In this paper, the general problem of real-time monitoring of a dynamic system is examined, with particular emphasis on the problem of detecting anomalous states which have not been modelled *a priori.* There is an implicit assumption that the system being monitored is not amenable to standard linear modelling so that the standard linear systems approach to fault detection described in the control theory literature [2, 3, 4] is not applicable — this is frequently the case with complex real-world systems. It will help the reader to imagine that the particular system being monitored is in fact a sub-system of an overall larger system: the primary purpose of monitoring the "subsystem" is to quickly determine if it in particular is the cause of

the problem, given that an overall anomaly has been detected. For example, we will later focus on the problem of health monitoring the pointing subsystem of a 34m DSN antenna — the overall larger system could be considered to be the complete ground station (antenna/microwave/recei ver/decoder) or indeed the entire communications link between the spacecraft and JPL.

2 The Hidden Markov Model Method for Online Monitoring

2.1 Basic Notation and Assumptions

Consider that we are monitoring a dynamic system for which observable measurements are available at discrete time intervals. Denote the observable d-dimensional random variable as Y where $y \in \mathcal{R}^d$ is a particular realisation of Y, Without loss of generality assume that the time interval is 1 so that at time t we have seen a sequence of such measurements, $\Phi^t = \{\underline{y}(t), \underline{y}(t-1), \ldots . \underline{y}(0)\}$, where Φ^t represents all the observed data up to time t, and $\underline{y}(t)$ is the observed data at time t.

Let Ω be a discrete random variable taking values in the set $\{\omega_1, \ldots . \omega_m\}$. Assume that the system at any time t is in one and only one of the m states, WI, , , , , ω_m. For fault diagnosis applications ω_1 is typically chosen to be the normal operating state of the system and the other states represent various possible fault conditions. The assumption that the states are mutually exclusive is equivalent to the "single-fault" assumption, namely that multiple faults do not occur simultaneously — this is a reasonable assumption in practice when fault conditions are relatively rare. The second assumption, that the set of fault states is exhaustive, is much more restrictive and rarely likely to be true in practice: we will return to this aspect of the problem later in the paper, but for the present we will assume that exhaustivity holds.

Now consider the observable Y in relation to the states. We will assume that the conditional probability of Y given a particular state is stationary, i.e.,

$$p(\mathbf{Y}(t_1) = \underline{y} | \Omega(t_1) = \omega_i) = p(\mathbf{Y}(t_2) = \underline{y} | \Omega(t_2) = \omega_i), \qquad \forall t_1, t_2, \qquad 1 \leq i \leq m.$$

For shorthand throughout the paper we will replace "$\Omega(t) = \omega_i$" by ω_i^t and "$\mathbf{Y(t)} = \underline{y}$" by $\underline{y}(t)$. Since the state-observable relation is not time-dependent, explicit reference to time t in this context will be omitted,

The observable $\underline{y}$ can be considered to be a probabilistic function of the underlying states, As an example, consider the case where the dynamic system being observed can be modelled in closed form, perhaps by a set of linear equations, In this case, the observable $\underline{y}$ could be a vector of residual prediction errors between the predicted system outputs at time t and the actual system outputs at time t. Thus, $\underline{y}$ is a characteristic indicator of whether the system is remaining with its normal state or has switched to another state (one which is not well approximated by the closed form system model). We will examine in more detail later specific examples of observable and underlying states.

At this point we will assume that the prior probabilities of the states $p(\omega_i)$ are known and that either (a) the conditional probabilities of the observable given the states $p(\underline{y}|\omega_i)$ are known, or equivalently, (b) that the conditional probabilities of the states given the observable $p(\omega_i|\underline{y})$ are known. Since the states have been assumed to be mutually exclusive and exhaustive, the equivalence of knowing either (a) or (b) follows from Bayes' rule:

$$p(\omega_i|\underline{y}) = \frac{p(\underline{y}|\omega_i)p(\omega_i)}{\sum_{j=1}^{m} p(\underline{y}|\omega_j)p(\omega_j)}, \qquad 1 \leq i \leq m.$$

Section 2.3.2 will discuss how these conditional distributions may be estimated in practice.

Finally we will assume that the state variable $\Omega(t)$ is a stationary discrete-time first-order Markov process with transition matrix $\mathbf{A}$. Entry a_{ij} is the probability of the system being in state j at time t given that the system was in state i at time $t-1$, i.e.,

$$a_{ij} = p(\omega_j^t|\omega_i^{t-1}), \qquad \forall t, \qquad 1 \leq i,j \leq m. \tag{1}$$

$\Omega(t)$ need not necessarily be assumed to be a first-order Markov process: the algorithms and methodology described in the paper can be readily extended to higher-order and semi- Markov processes.

What we have described so far is a first-order discrete-time *hidden* Markov model — a diagrammatic representation is shown in Figure 1. The hidden aspect of the model reflects the fact that the states are not directly measurable themselves, but are indirectly observable via the $\underline{y}$'s which are a probabilistic function of the underlying Markov process. In general, the parameters of a specific model are often referred to generically as $\lambda = \{p(\omega_1), \ldots, p(\omega_m), p(\underline{y}|\omega_1), \ldots, p(\underline{y}|\omega_m), \mathbf{A}\}$.

2.2 Inference of the State Probabilities given the Observed Data

Given a particular hidden Markov model with m states and fixed parameters λ, one can use this model to infer the posterior probabilities of each of the hidden states at time t, *given* the observed data $\Phi^t = \{\underline{y}(t), \underline{y}(t-1), \ldots, \underline{y}(0)\}$ up to and including time t. The application might be an online system where we are observing the data in real-time and wish to generate an alarm if $p(\omega_1^t|\Phi^t)$ <0.5 (for example). The solution to this problem follows directly from the assumptions in the previous sections and an efficient algorithm for recursively computing these state estimates has been developed in the speech recognition literature (the "forward algorithm" — see Rabiner [5] for a tutorial overview). In particular we have that

$$p\left(\omega_j^t|\Phi^t\right) = \frac{1}{C^t} p\left(\underline{y}(t)|\omega_j^t\right) \sum_{i=1}^{m} a_{ij} p\left(\omega_i^{t-1}|\Phi^{t-1}\right), \qquad t \geq 1, 1 \leq j \leq m \tag{2}$$

where

$$p\left(\omega_j^1|\underline{y}(0)\right) = p(\omega_j), \qquad 1 \leq j \leq m.$$

and,

$$C^t = \sum_{j=1}^{m} \left[p\left(\underline{y}(t)|\omega_j^t\right) \sum_{i=1}^{m} a_{ij} p\left(\omega_i^{t-1}|\Phi^{t-1}\right) \right] \tag{3}$$

is just a normalizing constant to ensure that the state probability estimates at time t sum to 1. Note that these equations are usually written in terms of α variables in the speech recognition literature.

In a similar manner, at time t, one can calculate the probability of state j at time t – k given only the data which has occurred between time $t - k$ and time t, defined as:

$$\Gamma^{t-k} = \{\underline{y}(t), \underline{y}(t-1), \ldots, \underline{y}(t-k+1)\}.$$

Analogous to the previous recursion, but now conditioning on data which occurred *after* time $t - k$ one has

$$p\left(\omega_i^{t-k}|\Gamma^{t-k}\right) = \frac{1}{L^t} \sum_{j=1}^{m} a_{ij} p\left(\omega_j^{t-k+1}|\Gamma^{t-k+1}\right) p\left(\underline{y}(t-k+1)|\omega_j^{t-k+1}\right),$$

$$1 \leq k \leq t, \qquad 1 \leq i \leq m \tag{4}$$

where

$$p\left(\omega_i^t|\Gamma^t\right) = \frac{1}{m}$$

and L^t is a normalisation constant. This is known as the 'backward recursion algorithm" in speech-related work and is often written in terms of intermediate variables known as β's.

Thus, for a state which occurred in the past at time $t-k$, one can calculate the probability of that state given the **data** which occurred up to time $t-k$, and after $t-k$ to time t, by combining the $p(\omega_j^{t-k}|\Phi^{t-k})$ and $p(\omega_j^{t-k}|\Gamma^{t-k})$ in the following manner (see Rabiner [5] for proofs):

$$p\left(\omega_j^{t-k}|\Phi^t\right) = \frac{p\left(\omega_j^{t-k}|\Phi^{t-k}\right)p\left(\omega_j^{t-k}|\Gamma^{t-k}\right)}{\sum_{i=1}^{m} p\left(\omega_i^{t-k}|\Phi^t\right)p\left(\omega_i^{t-k}|\Gamma^{t-k}\right)},$$

$$1 \le k \le t, \qquad 1 \le j \le m. \tag{5}$$

Equations (2), (4), and (5) are the basic estimation equations for online monitoring.

2.3 HMM's applied to online monitoring

Given the above monitoring equations it remains to determine the Markov model structure and parameters in order to implement the method for a particular problem.

2.3.1 HMM transition matrix

The key point in the Markov monitoring approach (previously introduced in [6] and [7]) is that the transition probabilities are not estimated from the data (as in speech modeling), but rather are chosen *a priori* based on the long-term temporal characteristics of the system and prior knowledge concerning the system failure modes. Note that the use of this prior information is essential in applications of this nature in the sense that there is often no alternative practical method by which to estimate these probabilities. For example, in speech modelling one can use estimation algorithms (such as the Baum-Welch algorithm) to estimate the probabilities via maximum likelihood methods directly from data. In online monitoring however, sequences of data containing normal-fault transition information are not likely to be available. Hence, prior knowledge must be used to determine the probabilities in the matrix A.

In particular, the mean time spent in the normal state ω_1 of the model should be equal to the mean time between failure (MTBF) of the observed system. For a

first-order Markov system the expected length of time l spent in state ω_1, given that it starts in state ω_1, is

$$\begin{aligned} E[l] &= \sum_{n=1}^{\infty} n a_{11}^{n-1}(1 - a_{11}) \\ &= \frac{1}{1 - a_{11}} \\ &= \text{MTBF} \end{aligned}$$

Hence,

$$a_{11} = 1 - \frac{1}{\text{MTBF}}. \tag{6}$$

In practice the MTBF of a particular system is typically either available from databases of trouble reports or can be inferred from reliability design information (for a new system).

Given all, the a_{1i} terms (where $2 \leq i \leq m$) are chosen based on the relative likelihoods of the known fault states and are subject to the constraint that $\sum_{i=2}^{m} a_{1i} = 1 - a_{11}$. Once again, these values are chosen based on prior knowledge about the relative probabilities of the particular fault states in the model: this information again may be available from a database of trouble reports or can be inferred from first principles reliability analyses of the various fault states.

The remaining probabilities of the form a_{ij}, $i \neq 1$, are dependent on whether the faults under consideration are transient in nature or are such that the only way to return to the normal state is by shutting down the system and repairing the fault. Hence, the exact nature of the HMM matrix is highly dependent on the particular operational scenario under which the model is being used. A more detailed discussion on the design of HMM parameters is given in [7].

2.3.2 Estimation of the Observable-State Conditional Dependencies

Earlier we showed that the problem of estimating the probability of any state at time t based on a particular sequence of observed data (either before, or after t, or both) can be expressed in recursive form. A central component in the recursion is the calculation of the *instantaneous* observable-state conditional probability, $p(\underline{y}|\omega_j)$, namely the probability of observing the data $\underline{y}$ supposing that the system is in state ω_j at time t.

Consider the problem of estimating $p(\underline{y}|\omega_j)$ for a specific state ω_j. The density function can be either chosen directly in the form of an explicit parametric model (for example, a multi-variate Gaussian model) or could be estimated implicitly from data via non-parametric methods such as multi-variate kernel density estimation [8, 9]. In the parametric model case, the parameters of the model can in turn either be chosen based on prior knowledge of the system characteristics under fault and normal conditions, or can be estimated from data. The data-based estimation procedures (whether parameter fitting for a specific model or the entirely non-parametric approach) requires that observed data is available for state w_j. This is usually not a problem for the normal state ω_1. Indeed data can often be generated for common fault states which are easy to emulate in hardware — later we will describe results using this data-dependent approach.

It is worth noting that the dimensionality of the observable variable Y can have a distinct effect on one's ability to estimate the functional form of the model $p(\underline{y}|\omega_j)$. The larger the dimensionality the more parameters must be estimated (in the parametric case) and the sparser the data (in the non-parametric case), hence, the "curse of dimensionality" may apply. One approach to alleviating this problem is to use Bayes' rule and work with the *posterior* or *discriminative* probabilities of the states given the observable. It is straightforward to show that the recursive state estimation equations derived earlier (see Equation (2)) can equally well be written in *discriminative* form:

$$p\left(\omega_j^t|\Phi^t\right) = \frac{1}{H^t}\frac{p(\omega_j^t|\underline{y}(t))}{p(\omega_j)}\sum_{i=1}^{m} a_{ij}p\left(\omega_i^{t-1}|\Phi^{t-1}\right), \qquad t \geq 1, 1 \leq j \leq m \tag{7}$$

where

$$p(\omega_j^1|\underline{y}(0)) = p(\omega_j), \qquad 1 \leq j \leq m,$$

H^t is a normalization constant as before to ensure that the state probability estimates at time t sum to unity, and $p(\omega_j)$ is the prior probability of state j. This is equivalent to the previously derived recursion relation by virtue of Bayes' rule:

$$\frac{p(\omega_j^t|\underline{y}(t))}{p(\omega_j)} = \frac{p(\underline{y}(t)|\omega_j^t)}{p(\underline{y}(t))}.$$

In a manner exactly analogous to that of Equations *(4)* and (5), one can also derive equivalent recursion relationships for using all the observed data to time t, Φ^t, to estimate the probability of a state at time $t-k$.

The advantage of this discriminative formalism is that estimation of the discriminative probabilities tends to scale better (in terms of estimation) with dimensionality (of the observable) and is less sensitive to particular distributional assumptions about the observable, than the alternative approach of modelling $p(\underline{y}|\omega_i)$ directly. Candidates for estimating such discriminative probabilities directly from data include logisitic discrimination [1 O] and feedforward neural network models [1 1].

3 Previous Results on Antenna Monitoring

In previous work [6, 7] we have described the application of the Markov monitoring approach to online failure detection in antenna pointing systems which are used to steer large 34 and 70 meter ground antennas of the DSN during tracking of various deep-space spacecraft. Figure 2 shows a block diagram of a typical signal path from a spacecraft back to mission control at JPL. The ground antenna is but one link in a complex chain of communications and signal processing equipment. When a problem occurs, it is important to quickly isolate the cause of the problem whether it is within the ground station, in the spacecraft, or elsewhere. Because of the complexity of the link, it is not unusual that problems can not be tracked down in real-time, resulting in loss of telemetry data and early shut-down of the track. Furthermore, post-event trouble-shooting may not be able to accurately recreate the exact problem symptoms, resulting in either ignoring the problem until it becomes so severe the station cannot track a spacecraft or misdiagnosing the problem and perhaps replacing the wrong hardware component. The antennas are probably the single most difficult piece of equipment to monitor accurately. Standard linear models are not particularly effective in practice due to the presence of various non- linearities in the mechanical gears and bearings and environmental influences such as wind gusts.

An experiment was carried out whereby hardware faults were introduced into the pointing system of a 34 meter antenna at the Goldstone antenna complex in a controlled manner, Separate sets of data were collected on two different days; a training set to train the model and a test set to evaluate its performance on unseen data. For each data set, data from a variety of sensors (motor current, tachometer, counter torque readings sampled at 50 Hz) were recorded under known conditions. The four known conditions corresponded to (a) normal conditions, (b) a noisy tachometer

(corresponding to brush or bearing wear), (c) complete failure of a tachometer, and (d) a short-circuit in an amplifier.

The input feature vector $\underline{y}$ consisted of 8 autoregressive-exogenous (ARX) coefficients and 4 standard deviation measurements, resulting in a 12 dimensional feature space. The features were estimated from non-overlapping sequential blocks of sensor data where each block consisted of 200 consecutive sample vectors, The ARX model used the rate command to the system as the driving term and the motor current as the response. A model of the form

$$x(t) + \sum_{i=1}^{p} a_i x(t - i) = \sum_{j=1}^{q} b_j u(t - j) + e(t)$$

was used where $p = 6$, $q = 2$, $e(t)$ is an additive white noise process, and $x(t)$ and $u(t)$ are the motor current and rate command respectively. The feature coefficients, a_i and b_j, are completely re-estimated (via standard least-squares estimation) for each window

Hence, from the original time series data sampled at 50 Hz, a 12-dimensional feature vector is produced every 4 seconds. This is the observable feature vector $\underline{y}$ for the HMM. The training data consisted of 75 such feature vectors for each of the fault states, and 225 vectors for the normal state. We experimented with two different methods for estimating the state-observable conditional dependencies. First we used a 3-component Gaussian mixture model for each of the 4 states to directly model $p(\underline{y}|\omega_j)$, i.e., 4 such models were used. The mixture models were fitted via a standard iterative maximum likelihood procedure known as the EM algorithm [12], The component densities were constrained to have diagonal covariance matrices to reduce the number of parameters which needed to be estimated.

The second model used was a discriminative model (i.e., it modelled $p(\omega_j|\underline{y})$ directly), in particular, a feedforward single hidden layer neural network. The model had 12 inputs, 8 hidden units, and 4 output units (one for each class) and used non-linear sigmoidal activation functions in both the hidden and output units (the sigmoid is of the form $f(x) = 1/(1 + exp(-x))$). The network weights were estimated ("trained") via a conjugate-gradient variant of the backpropagation algorithm to minimise the empirical log-likelihood [13, 14]. Under appropriate theoretical conditions it has been shown elsewhere [15, 16] that the outputs of such a trained network converge to the true *a posteriori* probabilities as the training sample size gets large. Hence, although

in practice one only has a finite training sample size, and the architecture of one's network model may be biased, it is reasonable to interpret the network outputs as estimates of the posterior probabilities of the states given the observable.

Results with both the mixture and neural models (without any HMM component) are shown in Figures 3a and 3b respectively. The vertical axes represent various posterior probability estimates of the states $p(\omega_i|\underline{y})$, while the horizontal axis represents time in minutes, The system begins in the noisy tachometer fault state, switches to the failed tachometer state, switches back to the normal state, switches to the amplifier fault state, and finally returns to normal. Note that neither model is particularly accurate in tracking the states and that there is a tendency to switch around between states. This is a direct consequence of ignoring the temporal aspect of the problem, i.e., ignoring the fact that the fault and normal states do not occur randomly in time but tend to *persist.* It can also be seen in Figure 3a that the mixture model's probability estimates tend to be near either O or 1 indicating sharp decision boundaries and perhaps overfitting (the widths of the component Gaussian mixtures may be too small). The neural network model on the other hand provides smoother estimates.

A simple HMM was designed to model temporal context. All states were chosen to be equally likely and the probability of changing to a different state was assigned as 0.01, resulting in a probability of remaining in the same state of 0,97 (this corresponds to an effective expected duration in any one state of 2 minutes for our simulated problem). Note that in a practical situation the transition probabilities would tend to be much closer to O or 1 in order to model realistic MTBF's. The depth of the backward recursion (k in equations (4) and (5)) was set to 5 time steps: empirically it was found that setting k any greater than this resulted in no change in the estimates. Thus there is a latency of 5 x 4 = 20 seconds before the systems produces its estimate of the state probabilities for time t.

Figures 4a and 4b show the results obtained when this Markov transition matrix was used to correlate the estimates from Figures 3a and 3b, the mixture model and neural network state estimates, respectively. There is some improvement gained by introducing the HMM for the mixture model, for example, the number of false alarms (fault probabilities greater than 0.5 when the true state is normal) is somewhat reduced, However, overall the quality of the state probability estimates produced by the mixture model are poor enough that adding temporal context (in the form of the

HMM) does not significantly improve the overall quality of the model's estimates.

For the neural-HMM combination (Figure 4b), however, there is a dramatic improvement in performance. Apart from some mislabeling of the tachometer noise state, the model tracks the states almost perfectly. The lack of stability evident in Figure 3b (without the HMM) is completely removed, i.e., the introduction of appropriate temporal context removes any ambiguity about the identity of the states. Of the 4 responses plotted in Figures 3 and 4, due to its combination of low false rate and accurate detection, the neural-HMM combination in Figure 4b is the only one which is accurate enough for actual operational use in DSN ground stations.

4 Detecting Unknown States

4.1 Discriminative and Generative models

The assumption that there are m known mutually exclusive and exhaustive states (or "classes") of the system, o_1, ..., ω_m bears further investigation. The "mutually exclusive" assumption is reasonable in many applications where multiple simultaneous failures are highly unlikely. However, the exhaustive assumption is somewhat unrealistic. In particular, for fault detection in a complex system composed of hundreds of thousands of components, there are a myriad of possible fault conditions which might occur. The probability of occurrence of any single condition is very small, nonetheless there is a significant probability that at least one of these conditions will occur over some finite time. As an example, for the antenna application, there are a few well-known faults (such as tachometer failures) which occur with regularity and can be assigned specific fault states in advance; however, it is not practical to assign states to all the other minor faults which might occur.

Hence, the question must be asked as to whether or not a discriminant classifier trained to distinguish data from m states, can identify data from a different, or *novel* state. The answer lies in a simple application of Bayes' rule. If the classifier is a pure *discriminant*, i.e., it directly models the posterior probability $p(\omega_i|y)$, then it is implicitly relying on the assumption of exhaustivity and cannot in principle detect novel data. A good example of this type of classifier is a feedforward neural network using sigmoidal activation functions. Essentially, if one gives such a trained network new data which is far away from the training data in the feature space, it will produce

a near certain classification decision for one of the existing classes because of the semi-global nature of the sigmoid model [17].

On the other hand, a *generative* model directly models the data in the feature space conditioned on each class, $p(y|\omega_i)$, and then determines posterior class probabilities by application of Bayes' rule. Examples of generative classifiers include parametric models such as Gaussian classifiers and memory-based methods such as kernel density estimators and near neighbour models. Generative models are by nature well suited to online adaptation, in particular, adaptation of the structure of the model such as the inclusion of a new class. Conversely, discriminant models are by nature difficult to adapt online. However, there is a trade-off: because generative models typically are doing more modelling than just searching for a decision boundary, they can be less efficient (than discriminant methods) in their use of the data, For example, generative models typically scale poorly with input dimensionality for fixed training sample size and tend to be less robust to mismatches in terms of distributional assumptions about the observed data [10],

4.2 Hybrid Models with an "Unknown" State

To alleviate the fact that discriminative models may not detect novelty, but that generative models may have poor discriminative capabilities in many dimensions, we propose a hybrid classifier which uses *both* a generative and discriminative classifier in parallel. The model is based on the simple idea of adding an "m+ 1 th" state to the model to cover 'all other possible states" not accounted for by the known m states and modifying the probabilistic updating equations appropriately.

Let the symbol $\omega_{\{1,\ldots,m\}}$ denote the event that the true system state is one of the known states, and let $p(\omega_{\{1,\ldots,m\}}|\underline{y})$ be the posterior probability that the data is from the known state given the observed data $\underline{y}$. Hence, one can estimate the true posterior probability of individual known states as

$$p(\omega_i|\underline{y}) = p_d(\omega_i|\underline{y}, \omega_{\{1,\ldots,m\}})p(\omega_{\{1,\ldots,m\}}|\underline{y}),\ 1 \leq i \leq m \tag{8}$$

where $p_d(\omega_i|\underline{y},\omega_{\{1,\ldots,m\}})$ is the posterior probability estimate of state i as provided by a discriminative model (such as the neural network described earlier). Thus, the posterior estimates of the hybrid classifier are conditioned on whether or not the data comes from one of the m known classes.

The calculation of $p(\omega_{\{1,\ldots,m\}}|\underline{y})$ can be obtained as follows. Assume for the moment that $p(\underline{y}|\omega_{m+1})$, $p(\omega_{m+1})$, and $p(\underline{y}|\omega_{\{1,\ldots,m\}})$ are all known. Thus we have by Bayes' rule that

$$p(\omega_{\{1,\ldots,m\}}|\underline{y}) = \frac{p(\underline{y}|\omega_{\{1,\ldots,m\}})\,p(\omega_{\{1,\ldots,m\}})}{p(\underline{y}|\omega_{m+1})p(\omega_{m+1}) + p(\underline{y}|\omega_{\{1,\ldots,m\}})\sum_i^m p(\omega_i)} \quad (9)$$

and of course

$$p(\omega_{m+1}|\underline{y}) = 1 - p(\omega_{\{1,\ldots,m\}}|\underline{y})$$

Since by definition $p(\omega_{\{1,\ldots,m\}}) = 1 - p(\omega_{m+1})$, we must define a prior probability of being in an unknown state. This is based on the designer's prior belief of how often the system will be in this unknown state — a practical choice is that the system is at least as likely to be in an unknown failure state as any of the known failure states.

The $p(\underline{y}|\omega_{\{1,\ldots,m\}})$ term in Equation (9) is provided directly by the generative model, i.e., whereas the discriminative model provides direct estimates of $p_d(\omega_i|\underline{y}, \omega_{\{1,\ldots,m\}})$, the generative model provides estimates of the data given the states — hence, the notion of running a discriminative and generative model in parallel. The generative model need not necessarily use the full dimensionality of the input space, particularly if the dimensionality is large. A mixture model of the form

$$p(\underline{y}|\omega_{\{1,\ldots,m\}}) = \sum_{i=1}^{m} f(\underline{y}|\omega_i)p(\omega_i) \quad (10)$$

is a natural choice, where in turn, the individual $f(\underline{y}|\omega_i)$ components in the mixture could be chosen as unimodal parametric models (such as Gaussian), or mixture models (thus giving a hierarchical mixture model overall), or nonparametric kernel density estimates, The hierarchical mixture model provides a reasonable trade-off between model flexibility and complexity in practice,

Finally, the density of the observable data given the *unknown* state, $p(\underline{y}|\omega_{m+1})$, must be defined in Equation (9), This requires the use of non-informative Bayesian priors for $p(\underline{y}|\omega_{m+1})$ over a bounded space of feature values, i.e., the maximum entropy choice of priors which reflect the least amount of prior information. Typically this requires that we have some knowledge of upper and lower bounds on the possible feature values — with no prior knowledge on the nature of unknown faults a uniform density over this space is usually appropriate. Note that the stronger the bounds which can be placed on the features, the narrower the prior density, and the

better the ability of the overall model to detect novelty. If we only have very weak prior information (very wide bounds), this will translate into a weaker criterion for accepting points which belong to the unknown category.

4.3 Hybrid Models within a HMM context

The extension of the hybrid model to a HMM is straightforward, The number of states is now $m + 1$ instead of m, i.e., the 'unknown' state is explicitly modelled within the HMM. The necessary state-observable probabilities (as required for the HMM recursion) are calculated directly from Equations (8) and (9), i.e., the parallel generative/discriminative models are combined to provide estimates of both $p(\omega_i|\underline{y}(t)), 1 \leq i \leq m$ and $p(\omega_{m+1}|y(t))$ at each time instant. The transition probabilities between the known and transient states must be consistent with our choice of prior probability $p(\omega_{m+1})$, i.e., the relative likelihood of an unknown state. In addition, the self-transition probability of the unknown state reflects our prior belief concerning the typical duration of such states — in practice, multiple unknown states with different self-transition probabilities could be used to represent a range of possible conditions from short transient faults to longer-term "hard" failures.

5 Experimental Results on Detecting Novel States

5.1 Constructing Hybrid models in Practice

To test the hybrid model the following experiment was conducted using the same data as described in Section 3. In the training phase of the model the training data for one of the known states was omitted entirely, i.e., the model was trained on 3 states instead of 4. Then, using the approach of Section 4, a hybrid model to detect unknown states was constructed in the following manner:

- A discriminative classifier (the same feedforward neural network as described in Section 3) was trained using all 12 features to provide estimates of $p_d(\omega_i|\underline{y}, \omega_{\{1,\ldots,m\}}), 1 \leq i \leq 3$.
- A generative model consisting of a hierarchical mixture model (Equation (10)) was fitted to the training data using only 3 features derived from the motor cur-

rent signal; the two coefficients of a second-order autoregressive model (AR(2)), ϕ_1 and ϕ_2, and the standard deviation, σ. Hence, $\underline{y} = (\phi_1, \phi_2, \sigma)$. These features were chosen empirically on the basis of their being the most informative about changes in the system state — using all 12 features would not have been practical given that only 75 data points were available for the fault states. The individual mixture components were themselves a mixture of 3 Gaussian components with diagonal covariance matrices — the mixture weights and component parameters were estimated via the EM algorithm.

- The prior probabilities of the 3 known states and the unknown state were set uniformly, i.e., to 0.25,

- A 4-state HMM was designed; 3 known states plus 1 unknown state, all with equal prior probability and, as before, with probabilities of 0.01 of changing to any another state. k = 5 was again used as the backward recursion depth.

Finally, the prior density of the data given the unknown state, $p(\underline{y}|\omega_{m+1})$, was defined as follows. First consider the AR parameters ϕ_1 and ϕ_2. In accordance with standard time series theory, if the estimated process (as represented by the two coefficients) is to be stationary, then the coefficients must obey the following restrictions [18]:

$$\begin{aligned} \phi_1 + \phi_2 &< 1, \\ \phi_2 - \phi_1 &< 1, \\ -1 < \phi_1 &< 1. \end{aligned} \tag{11}$$

It will be assumed that the estimated coefficients are in fact stationary. Equation (11) thus provides bounds on the possible parameter values. A uniform density is specified over all such allowable values of ϕ_1 and ϕ_2. Of course, this does not allow for the fact that in practice (and in particular for fault conditions) there is no guarantee that the estimated coefficients will obey these bounds. The following approach is adopted: if the estimated coefficients lie outside the bounds of the stationary region then the probability of the unknown state $p(\omega_{m+1}|\underline{y})$ is set to 1.

The third parameter, the standard deviation of the voltage from the Hall effect sensor which measures motor current, is normally about 20mv under normal conditions. Based on experience from observing the motor current signal under a variety

of conditions it is estimated that under any fault condition the standard deviation should not exceed 1 volt. Hence, in the absence of any other prior information, a uniform density is placed on the standard deviation over this range O to 1 volt for a. This density is assumed independent of the AR(2) coefficient density, i e., the overall prior density is the product of the two.

5.2 Experimental Results

We conducted the experiment (of leaving a particular fault out of the training data set) twice; first the noisy tachometer fault was omitted, then the amplifier fault.

Figure 5a shows the response of the discriminative model alone (the feedforward neural network probability estimates coupled to a 3-state HMM) to the test data which contains an ‘unknown” fault, namely a noisy tachometer. The figure shows that the noisy tachometer fault is misclassified as a failed tachometer. Figure 5b shows the response when the hybrid model is run on the same data. Although there is some confusion between the unknown state and the failed tachometer state, the model clearly identifies that there is an unknown state present, i.e., the probability of an unknown state is usually close to 1 during the presence of the unknown noisy tachometer fault. Interestingly enough, the model also picks up a possible unknown transient which occurred during the simulation of the failed tachometer fault.

In the second experiment, the amplifier fault was removed from the training data but was present in the test set. Figure 6a shows the response of the discriminative 3-state HMM (no explicit model for unknown). Whereas in the previous experiment the unknown fault condition was merely classified incorrectly as being a different fault, here the unknown fault is almost completely missed, i.e., it is largely misclassified as being normal. This is exactly the type of problem which can arise when using discriminative models when the exhaustivity assumption is not met, namely, novel states can go undetected simply because they happen to sit on the “normal” side of the estimated decision boundaries. Figure 6b shows the response of the hybrid model to the same data. Now, although the unknown fault is not detected in its entirety, it is partially detected in the unknown category. As before, some of the tachometer fault data is also classified as being of the unknown variety, perhaps indicating various transients in the data.

6 Discussion of Related Work

The initial idea of treating the online monitoring of dynamic systems within the framework of a HMM appears to have been proposed independently by Heck and McClennan [19] and Smyth and Mellstrom [20]. Heck and McClennan [19] describe the application of the HMM approach to monitoring tool wear, with a particular application to drill-bit wear detection, In addition they propose the use of the HMM to provide predictive estimates of time-to-failure, a particularly useful approach when failures are of a costly and/or catastrophic nature, Ayanoglu [21] describes the use of HMM's for channel error detection in communications networks, Coast et al. [22] use HMM's for online monitoring and classification of cardiac arrhythmia signals, while Provan [23] also describes a novel application of essentially the same idea to the problem of medical diagnosis and treatment of acute abdominal pain. None of this work using hidden Markov monitoring dealt with the problem of detecting unknown states.

Previous work on novelty detection (within the context of classification/discrimination) has tended to focus on learning closed decision boundaries for the known classes (e.g., Hellstrom and Brinsley *[24])* — this approach is not likely to scale up well as the dimensionality of the input space increases, and may not be feasible at all with some classification models (such as neural network models with semi-global response functions such as sigmoids). An alternative approach, proposed by Dubuisson and Masson [25], is to use only a generative model in the model and thus detect outliers via distance thresholds, e.g., the distance from the mean in a parametric model, or the distance from the nearest training set point in near-neighbour or kernel models. The disadvantage of threshold-based methods lies in the selection of the thresholds themselves; without some hypothesis for data from the unknown state there is no principled way to choose such thresholds.

In contrast, the probabilistic model for detection of unknown states which we have proposed in this paper, while requiring significant prior knowledge on the part of the designer, is a more principled approach in that it makes quite explicit the various assumptions which surround the novelty detection issue. It forces the system designer to assign specific prior hypotheses to the prior density of the data conditioned on a novel event, Hence, the problem is framed explicitly in terms of comparing specific

hypotheses for known and unknown events.

7 Conclusion

In this paper the Markov monitoring approach has been extended to allow for detection of system states which were unknown *a priori*. This has significant impact for practical applications where novel events are far more likely than in a controlled laboratory environment. The proposed model was tested on data from a real-world fault detection application and clearly demonstrated its ability to detect previously unknown states — conversely, the model which assumed that unknown faults had a prior probability of zero either misclassified an unknown fault into another fault state or did not detect that any fault was present.

Acknowledgements

The author would like to thank Jeff Mellstrom for assistance in obtaining the experimental results. The idea of using both discriminative and generative classifiers was originally suggested by Richard Lippmann. The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and was supported in part by ARPA under grant number N00014-92-J-1860.

References

1. E. C. Posner and R. Stevens, "Deep Space communication — past, present and future," *IEEE Communications Magazine*, vol.22, no.5, 8-21, May 1984.

2. A. S. Willsky, 'A survey of design methods for failure detection in dynamic systems,' *Automatica, pp.601–611, 1976.*

3. R. Isermann, 'Process fault detection based on modeling and estimation methods — a survey,' *Automatica*, vol.20, 387-404, 1984.

4. P.M. Frank, 'Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy — a survey and some new results,' *Automatic,* vol.26, no.3, pp.459-474, 1990,

5. L. R. Rabiner, 'A tutorial on hidden Markov models and selected applications in speech recognition,' *Proc. IEEE*, vol.77, no.2, pp.257–286, February 1989.

6. P. Smyth and J. Mellstrom, 'Fault diagnosis of antenna pointing systems using hybrid neural networks and signal processing techniques,' in *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson, R. P. Lippmann (eds.), Morgan Kaufmann Publishers: San Mateo, CA, 1992, pp.667-674.

7. P. Smyth, 'Hidden Markov models for fault detection in dynamic systems,' *Pattern* Recognition, accepted for publication.

8. B. Silverman, Density *Estimation for Statistics and Data Analysis, New* York: Chapman and Hall, 1986.

9. A. J. Izenmann, 'Recent developments in nonparametric density estimation,' *J. Amer. Stat. Assoc.*, VO1.86, pp.205-224, March 1991.

10. J. A. Anderson, 'Logistic discrimination,' in *Handbook of Statistics, Volume 2: Classification, Pattern Recognition, and Reduction of Dimensionality,* P. R. Krishnaiah and L. N. Kanal (eds.), North Holland, Amsterdam, pp.169–191, 1982.

11. J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation,* Addison-Wesley, Redwood City: CA, 1991,

12. R. A. Redner and H. F. Walker, 'Mixture densities, maximum likelihood, and the EM algorithm,' *SIAM* Review, vol.26, no.2, pp. 195-239, April 1984.

13. E. Barnard and R. Cole, 'A neural net training program based on conjugate-gradient optimization,' Oregon Graduate Centre Technical Report No, CSE 89-014, Oregon, 1989.

14. M, J. D. Powell, 'Restart procedures for the conjugate gradient met hod,' *Mathematical Programming,* vol.12, pp.241–254, April 1977.

15. M. D. Richard and R. P. Lippmann, 'Neural network classifiers estimate Bayesian a posteriori probabilities,' *Neural Computation, 3(4),* pp.461–483, *1992.*

16. J. Miller, R. Goodman, and P. Smyth, 'On loss functions which minimize to conditional expected values and posterior probabilities,' *IEEE Transactions on Information Theory*, July 1993.

17. P. Smyth, 'Probability density estimation and local basis function neural networks,' in *Computational* **Learning** *Theory and Natural Learning Systems II*, T. Petsche, M. Kearns, S. Hanson, R. Rivest (eds.), Cambridge, MA: MIT Press, to appear, 1993,

18. G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control,* San Francisco, CA: Hold en- Day, 1970.

19. L. P. Heck and J. H. McClennan, 'Mechanical system monitoring using Hidden Markov models,' in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,* IEEE Press, New York, pp. 1697-1700, 1991.

20. P. Smyth and J. Mellstrom, 'Fault diagnosis of antenna pointing systems using hybrid neural networks and signal processing techniques,' in *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson, R. P. Lippmann (eds.), Morgan Kaufmann Publishers: San Mateo, CA, 1992, pp.667-674.

21. E. Ayanoglu, 'Robust and fast failure detection and prediction for fault tolerant communication networks," *Electronics Letters, 28(10),* pp.940-941, *1992.*

22. D. A. Coast, R. M. Stern, G. G. Cane, S. A. Briller, 'An approach to cardiac arrhythmia analysis using hidden Markov models,' *IEEE Trans. Biomedical Engineering,* vol.37, no.9, pp.826-836, 1990.

23. G. M. Provan, 'Modelling the dynamics of diagnosis and treatment using temporal influence diagrams,' in *Proceedings of the Third Workshop on Diagnosis,* pp.97-106, *1992.*

24. B. Hellstrom and J, Brinsley, 'Characterization of network responses to known, unknown and ambiguous inputs,' in *Neural Networks for Signal Processing III: Proceedings of the 1993 IEEE-SP Workshop, C.* A. Kamm, G. M. Kuhn, B.

Yoon, R. Chellappa, S.Y. Kung (eds.), IEEE Press, New York, pp.226–231, 1993.

25. B. Dubuisson and M. Masson, 'A statistical decision rule with incomplete knowledge about the classes,' *Pattern Recognition,* vol.26, no.1, pp.155-165, 1993.

Figure Captions for "Markov Monitoring with Unknown States"

- Figure 1: Adiagram representing the operation ofa3-state hidden Markov model, The observed data y is available at times $t-1$, t,. . . . The dotted lines represent the possible state transitions of the system between each time instant. The solid line represents one possible state sequence; in this case, the system is in state ω_1 at time $t-1$, remains in this state at time t, moves to state ω_3 at time $t+1$ and on to state ω_2 at time $t+2$. The inference problem is to infer the most likely states given only the observed data $\underline{y}(t-1), \underline{y}(t), \ldots$.

- Figure 2: A simplified diagram of the downlink between a spacecraft and mission control at the Jet Propulsion Laboratory. There are three DSN ground stations at remote locations in California, Spain, and Australia.

- Figure 3: Model responses on test data sequence *without* hidden Markov monitoring. The horizontal axis represents time in minutes, the vertical axes represent the model's estimates of the 4 possible system states: (a) 3-component Gaussian (diagonal covariance matrices) mixture model trained via EM algorithm, (b) single hidden layer (12 hidden units) feedforward neural network model,

- Figure 4: Model responses on test data sequence *with* hidden Markov monitoring, The horizontal axis represents time in minutes, the vertical axes represent the model's estimates of the 4 possible system states. The hidden Markov model used $a_{ii} = 0.97$ and $a_{ij} = 0.01$, $i \# j$. (a) 3-component Gaussian (diagonal covariance matrices) mixture model trained via the EM algorithm, (b) single hidden layer (12 hidden units) feedforward neural network model.

- Figure 5: Model responses on test data sequence *with* hidden Markov monitoring where the tachometer noise has been omitted from the training data set. The horizontal axis represents time in minutes, the vertical axes represent the model's estimates of the possible system states. The hidden Markov models used $a_{ii} = 0.98$ (3-state model) or $a_{ii} = 0.97$ (4-state model) and $a_{ij} = 0.01$, $i \# j$: (a) 3-state (no unknown state) single hidden layer (12 hidden units) feedforward neural network discriminative model with HMM (b) 4-state (3 known plus unknown state) hybrid generative/discriminative model with HMM, discriminative model as in (a), generative model is a 9-component Gaussian (diagonal covariance matrices) hierarchical mixture model trained via the EM algorithm.

- Figure 6: Model responses on test data sequence *with* hidden Markov monitoring where the amplifier fault has been omitted from the training data set. The horizontal axis represents time in minutes, the vertical axes represent the model's estimates of the possible system states. The hidden Markov models used $a_{ii} = 0.98$ (3-state model) or $a_{ii} = 0.97$ (4-state model) and $a_{ij} = 0.01$, i # *j*: (a) 3-state (no unknown state) single hidden layer (12 hidden units) feedforward neural network discriminative model with HMM (b) 4-state (3 known plus unknown state) hybrid generative/discriminative model with HMM, discriminative model as in (a), generative model is a 9-component Gaussian (diagonal covariance matrices) hierarchical mixture model trained via the EM algorithm.

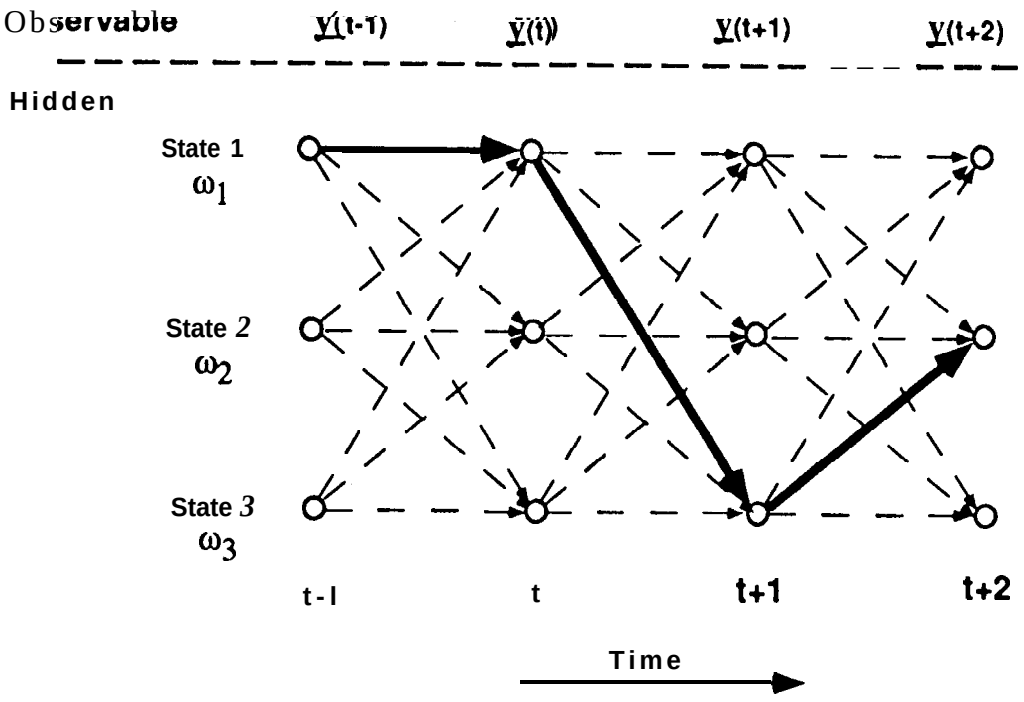

Observable
y(t-1)
y(t)
y(t+1)
y(t+2)
Hidden
State 1
ω1
State 2
ω2
State 3
ω3
t-1
t
t+1
t+2
Time

Figure 2.

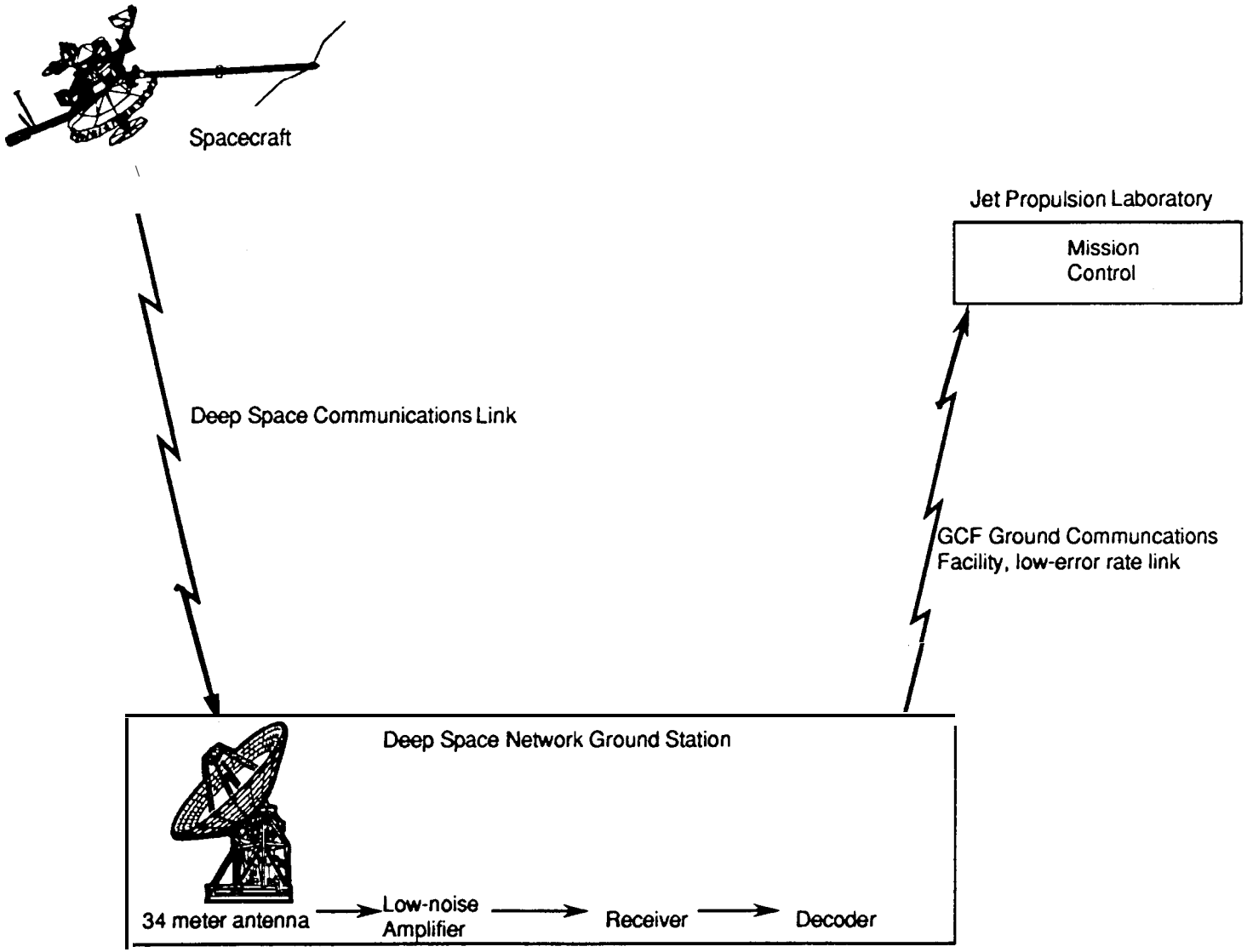
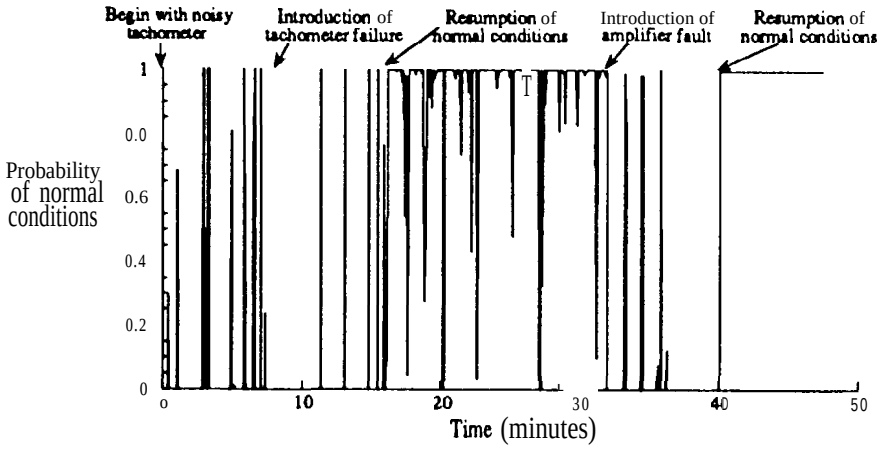
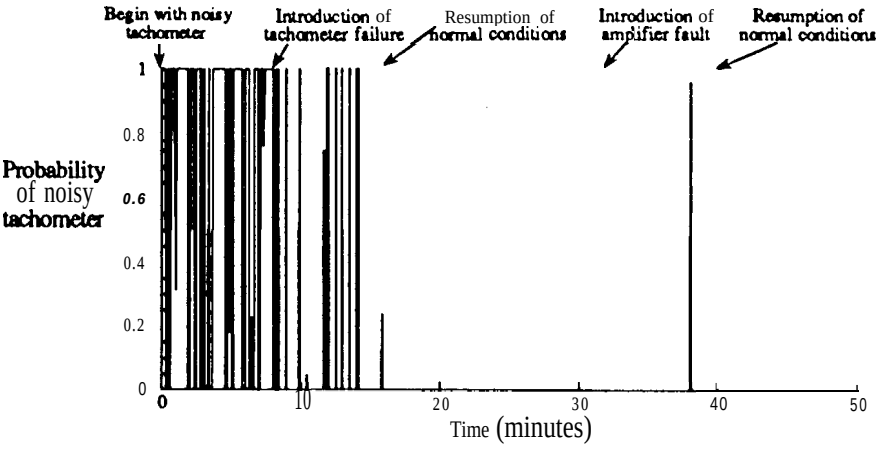
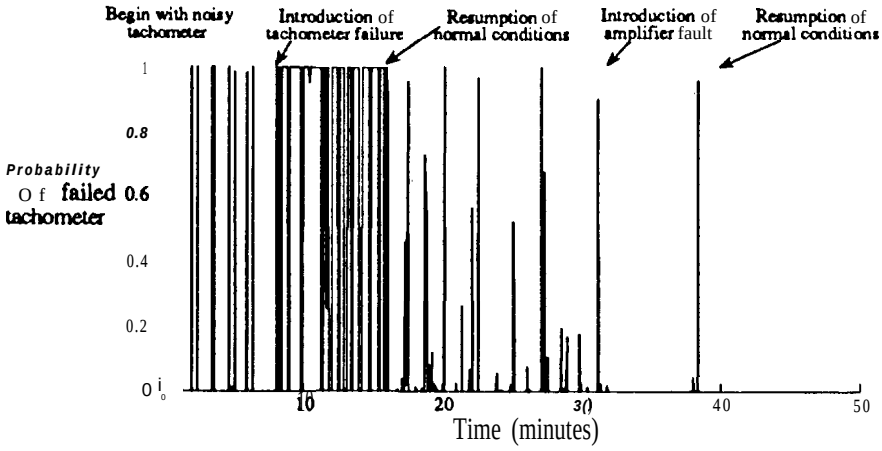

Spacecraft
Jet Propulsion Laboratory
Mission Control
Deep Space Communications Link
GCF Ground Communcations Facility, low-error rate link
Deep Space Network Ground Station
34 meter antenna
Low-noise Amplifier
Receiver
Decoder

Figure 3a


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of normal conditions
1
0.0
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of noisy tachometer
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of failed tachometer
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)

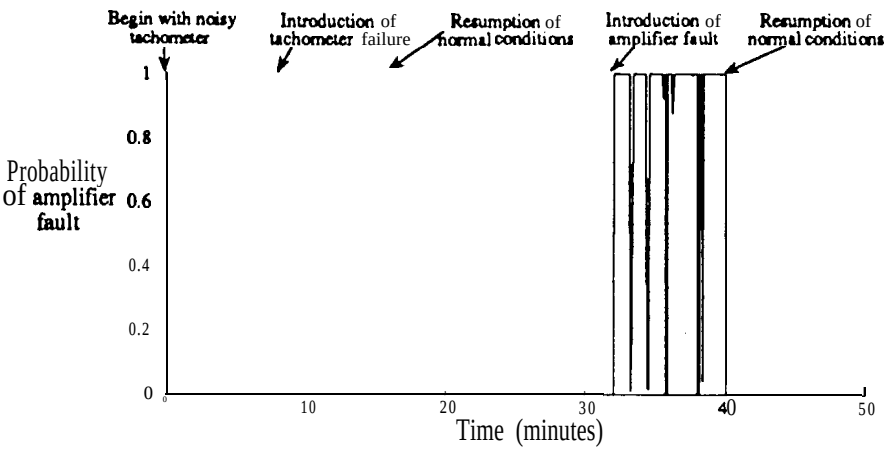
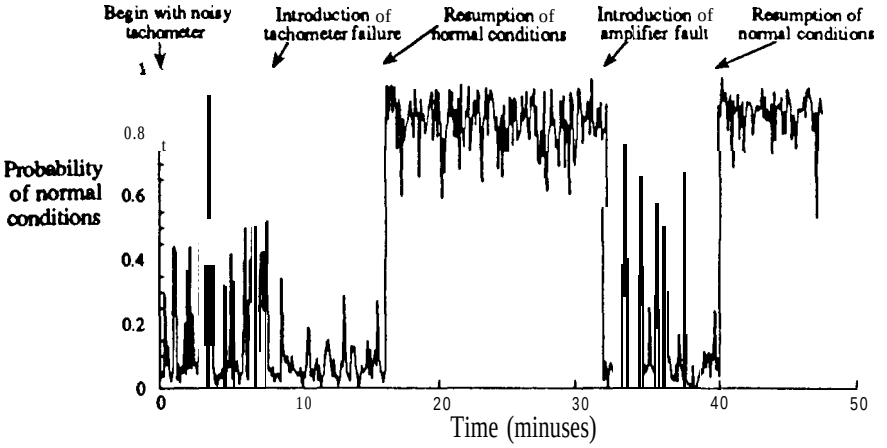
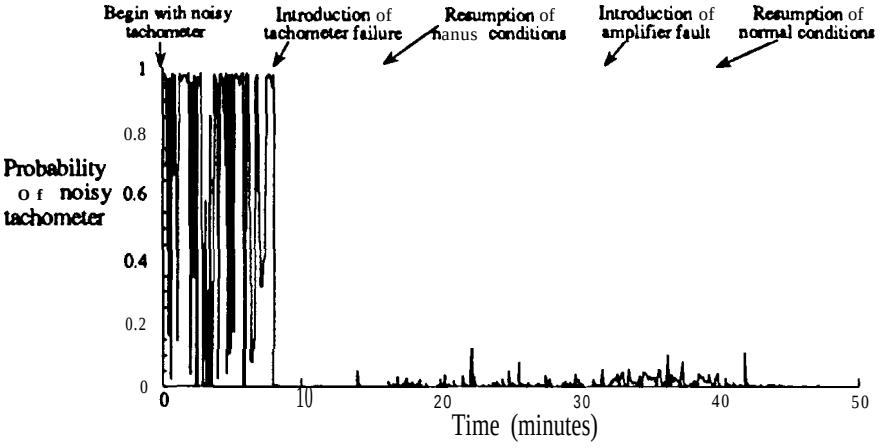
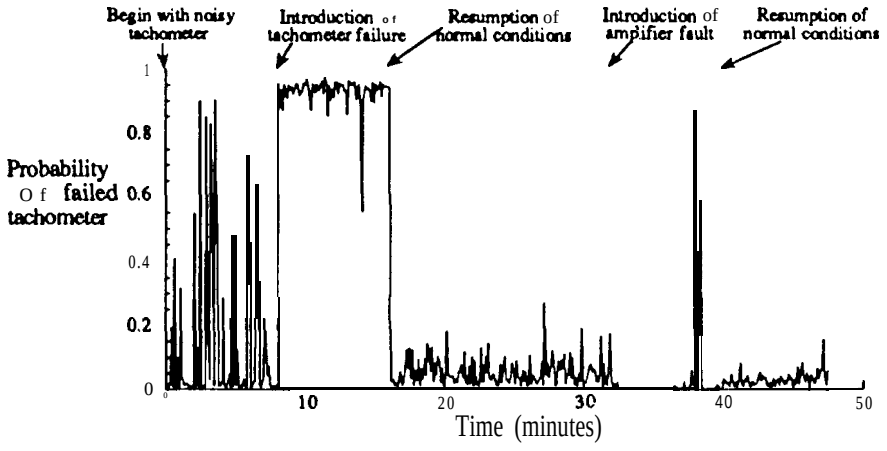

Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of amplifier fault
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)

Figure 36


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of normal conditions
0
0.2
0.4
0.6
0.8
1
0
10
20
30
40
50
Time (minuses)


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of hanus conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of noisy tachometer
0
0.2
0.4
0.6
0.8
1
0
10
20
30
40
50
Time (minutes)


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of failed tachometer
0
0.2
0.4
0.6
0.8
1
10
20
30
40
50
Time (minutes)

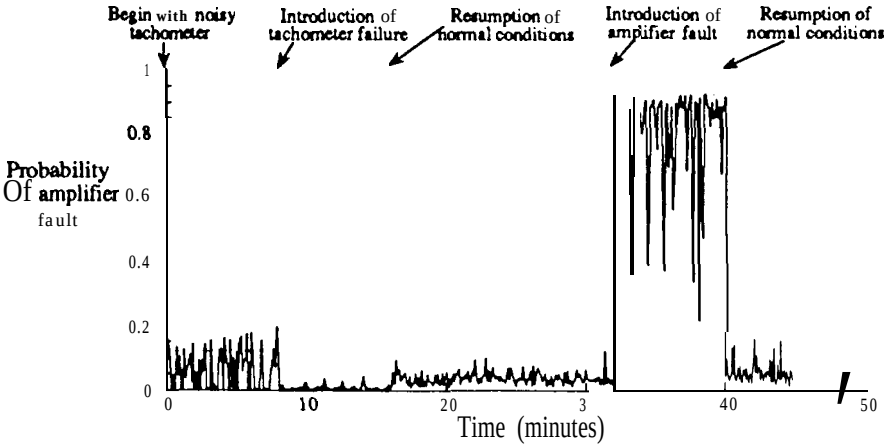
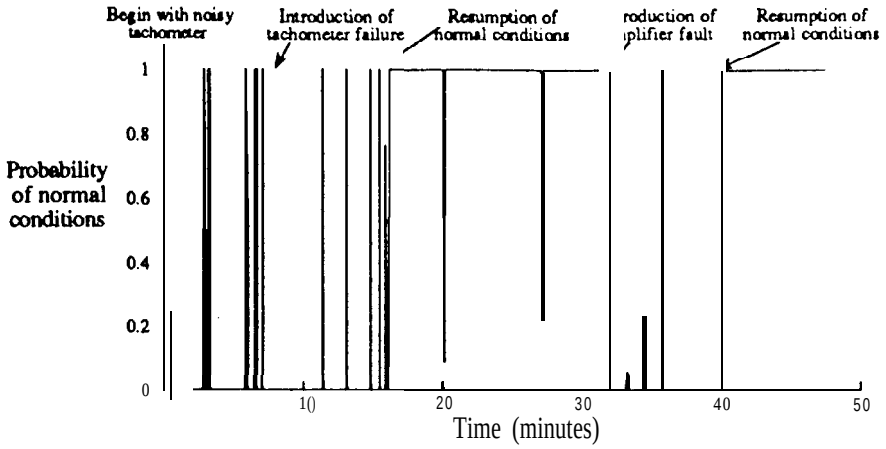
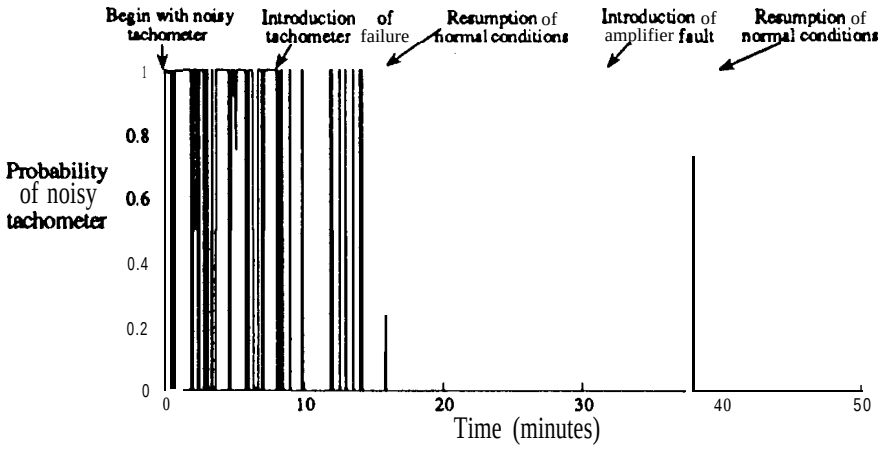
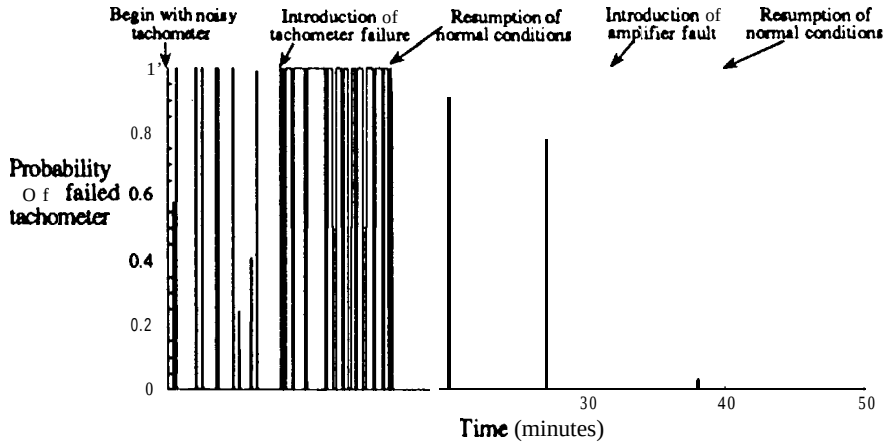

Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability Of amplifier fault
0
0.2
0.4
0.6
0.8
1
0
10
20
40
50
Time (minutes)

Figure 4a


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
roduction of plifier fault
Resumption of normal conditions
Probability of normal conditions
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of noisy tachometer
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability Of failed tachometer
1
0.8
0.6
0.4
0.2
0
30
40
50
Time (minutes)

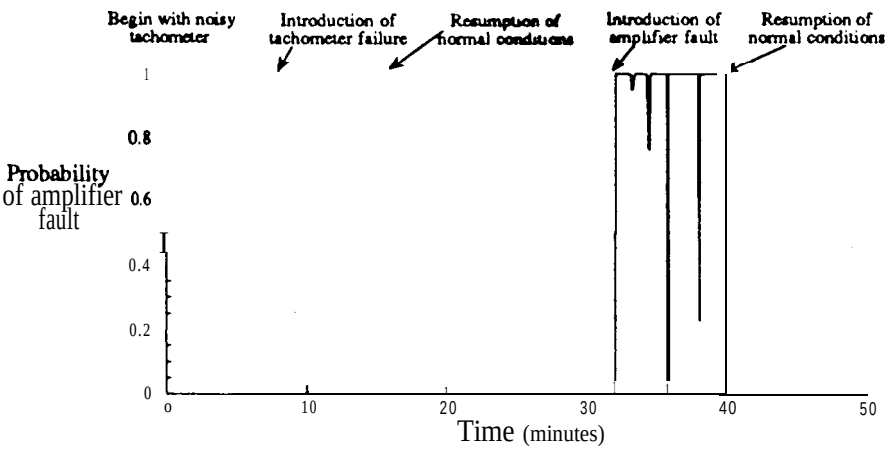
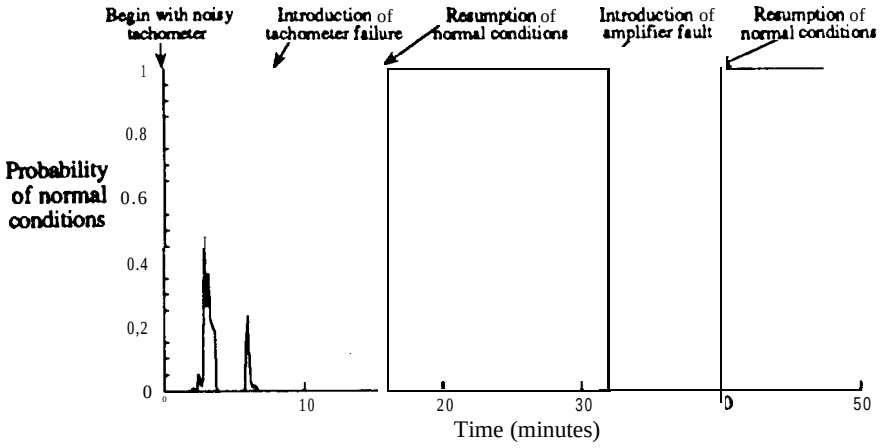
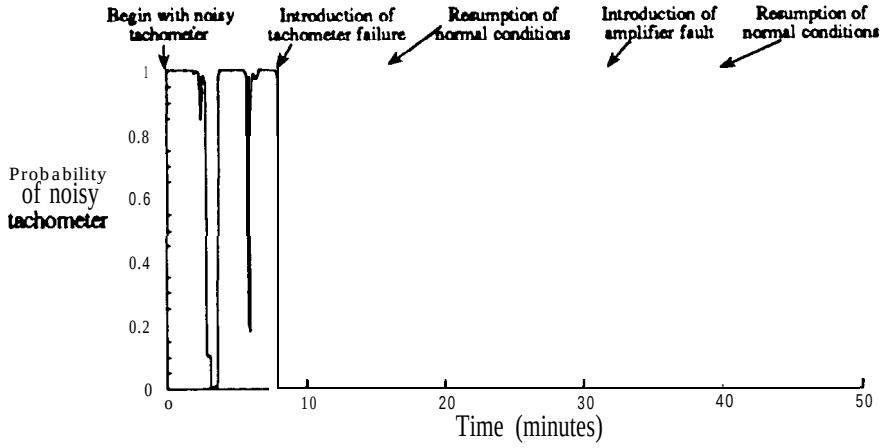
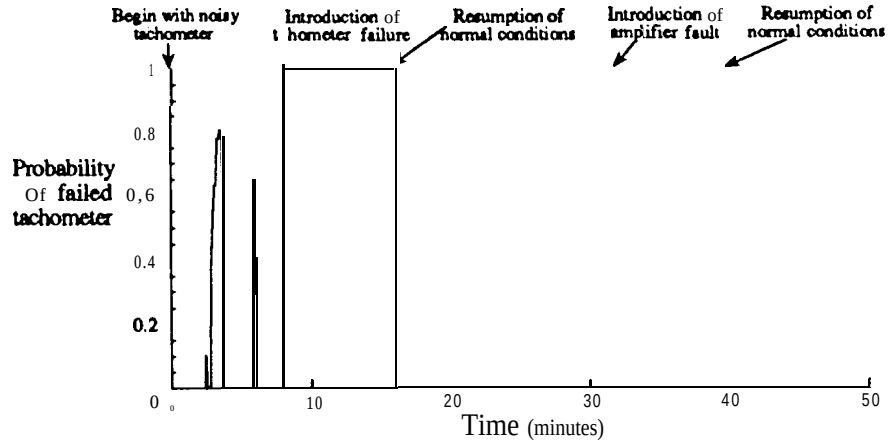

Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of amplifier fault
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)

Figure 4b


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of normal conditions
1
0.8
0.6
0.4
0,2
0
10
20
30
50
Time (minutes)


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of noisy tachometer
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)


Begin with noisy tachometer
Introduction of t hometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability Of failed tachometer
1
0.8
0,6
0.4
0.2
0
10
20
30
40
50
Time (minutes)

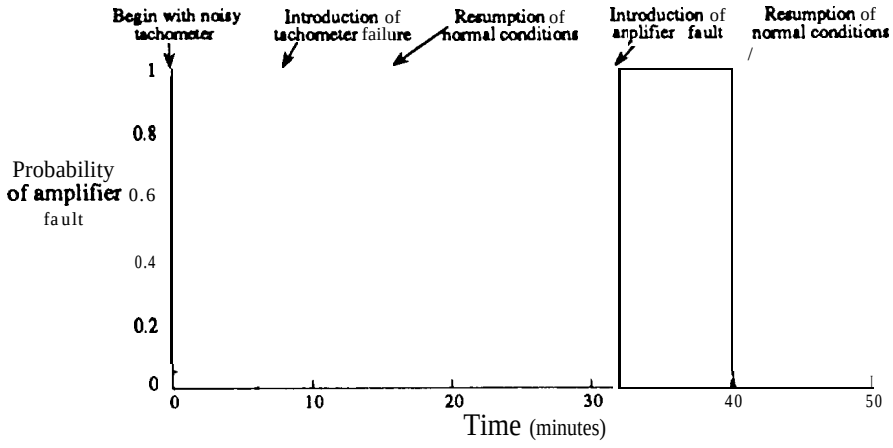

Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of amplifier fault
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)

Figure 5a

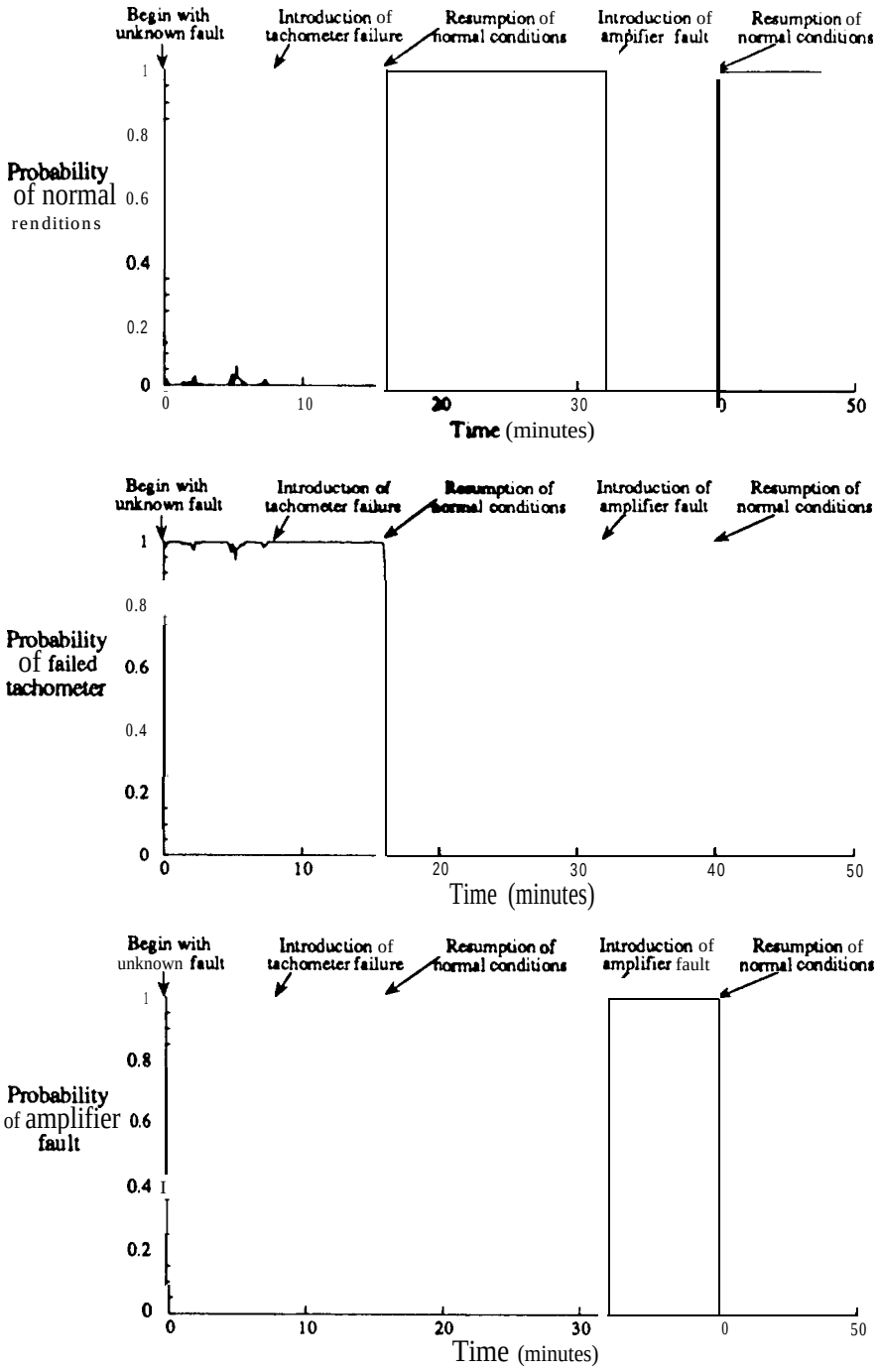

Begin with unknown fault
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of normal renditions
1
0.8
0.6
0.4
0.2
0
0
10
20
30
50
Time (minutes)
Begin with unknown fault
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of failed tachometer
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)
Begin with unknown fault
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of amplifier fault
1
0.8
0.6
0.4
0.2
0
0
10
20
30
0
50
Time (minutes)

Figure 56

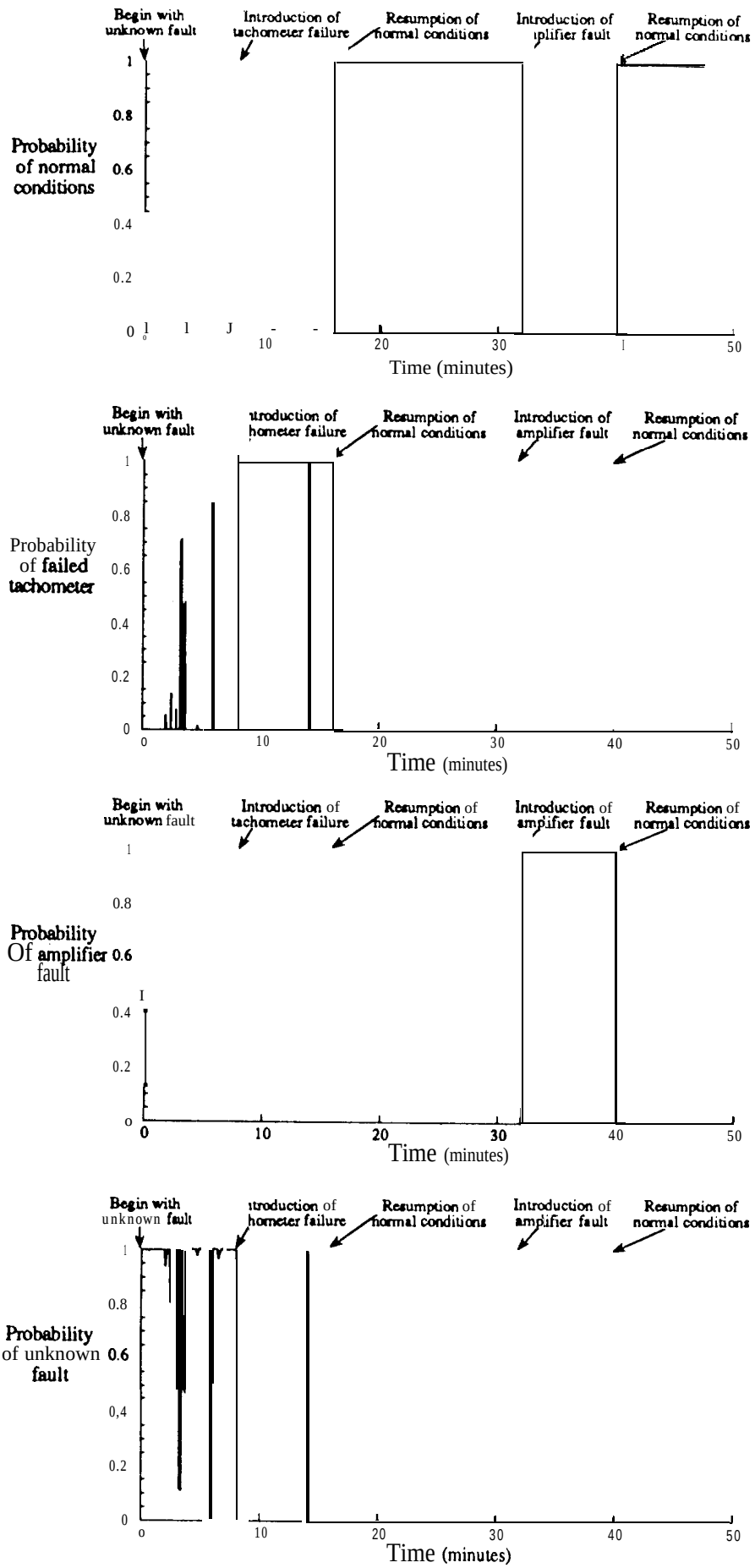

Begin with unknown fault
Introduction of tachometer failure
Resumption of normal conditions
Introduction of amplifier fault
Resumption of normal conditions
Probability of normal conditions
Probability of failed tachometer
Probability Of amplifier fault
Probability of unknown fault
Time (minutes)

Figure 6a

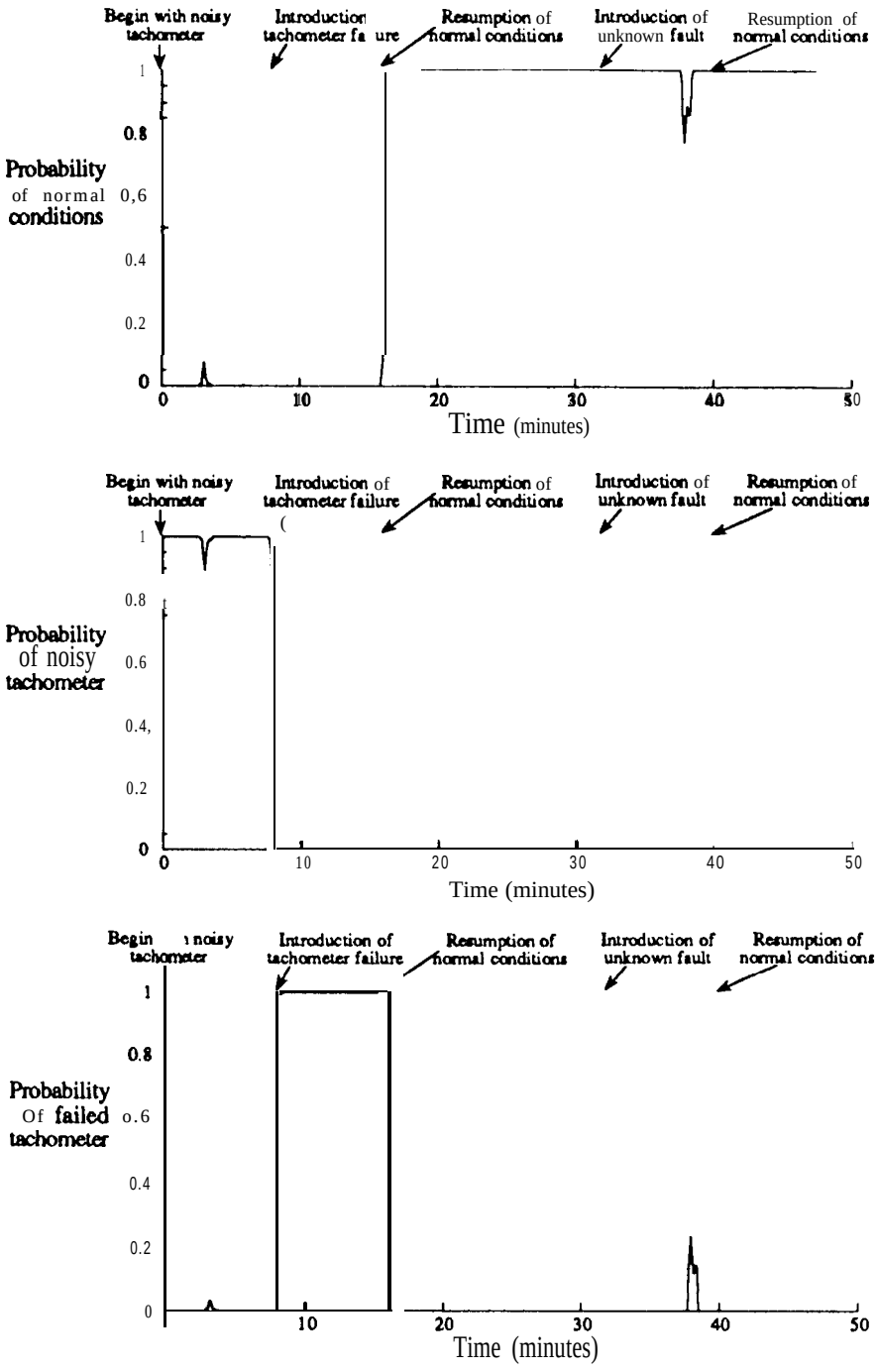
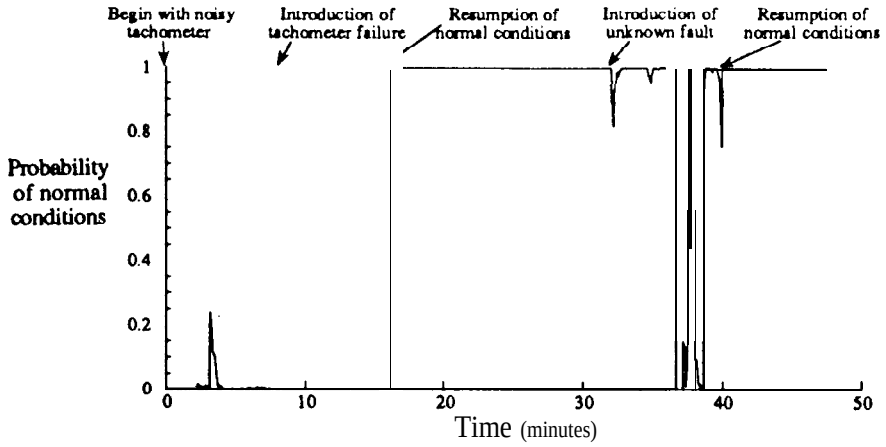
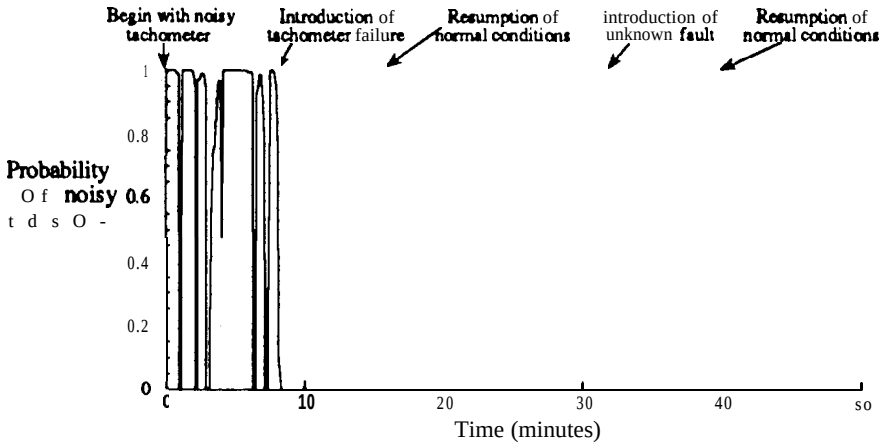
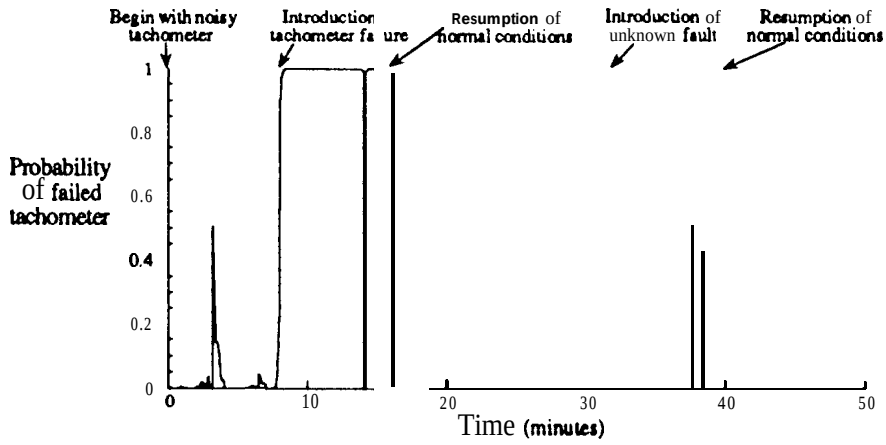
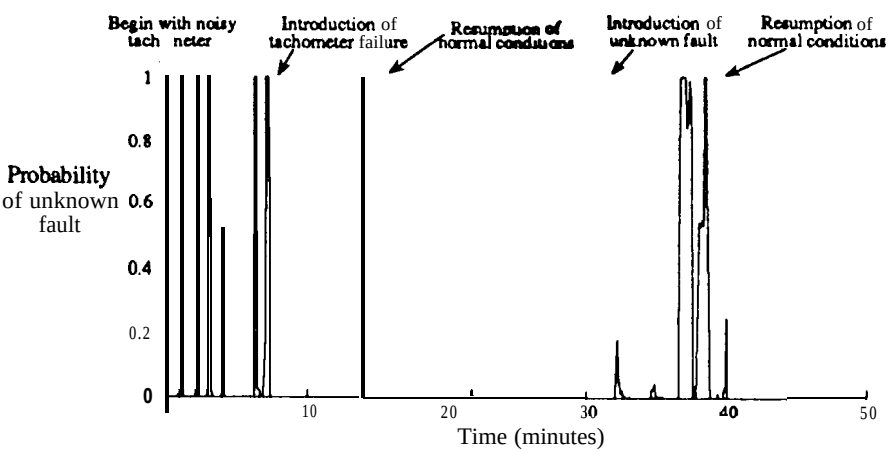

Begin with noisy tachometer
Introduction tachometer failure
Resumption of normal conditions
Introduction of unknown fault
Resumption of normal conditions
Probability of normal conditions
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)
Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of unknown fault
Resumption of normal conditions
Probability of noisy tachometer
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)
Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of unknown fault
Resumption of normal conditions
Probability Of failed tachometer
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)

Figure 66


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
Introduction of unknown fault
Resumption of normal conditions
Probability of normal conditions
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)


Begin with noisy tachometer
Introduction of tachometer failure
Resumption of normal conditions
introduction of unknown fault
Resumption of normal conditions
Probability
1
0.8
0.6
0.4
0.2
0
10
20
30
40
so
Time (minutes)


Begin with noisy tachometer
Introduction tachometer fa ure
Resumption of normal conditions
Introduction of unknown fault
Resumption of normal conditions
Probability of failed tachometer
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time (minutes)


Begin with noisy tach neter
Introduction of tachometer failure
Resumption of normal conditions
Introduction of unknown fault
Resumption of normal conditions
Probability of unknown fault
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Time (minutes)